NDB - Natural Stored Procedures

Stored procedures are user-written programs that are invoked by the SQL statement CALL and executed by DB2 in the SPAS (Stored Procedure Address Space), a separate address space allocated to (???) stored procedures.

This section covers the following topics:

- PARAMETER STYLE
- Writing a Natural Stored Procedure
- Security
- Example Stored Procedure

Related Topic:
Natural User-defined Functions in Statements and System Variables (Abschnitt???warum weg???)

PARAMETER STYLE

The PARAMETER STYLE identifies the linkage convention used to pass parameters to the DB2 (???) stored procedure.

Below are the PARAMETER STYLEs NDB provides for processing DB2 stored procedures:

- GENERAL and GENERAL WITH NULL
- DB2SQL

GENERAL and GENERAL WITH NULL

A Natural stored procedure defined with PARAMETER STYLE GENERAL only receives the user parameters specified.

A Natural stored procedure defined with PARAMETER STYLE GENERAL WITH NULL receives the user parameters specified and, additionally, a NULL indicator array that contains one NULL indicator for each user parameter.

Natural stored procedures defined with PARAMETER STYLE (GENERAL or GENERAL WITH NULL), require that the definition of the stored procedure within the DB2 catalog includes one additional parameter of the type VARCHAR in front of the user parameters of the stored procedure.

This parameter in front of the parameters is the STCB (Stored Procedure Control Block); see also the CALLDBPROC statement in Natural SQL Statements.

The STCB (???) contains information used by (???) the Natural stored procedures executed by the NDB server stub, such as the library and the subprogram to be invoked. It also contains the format descriptions of the parameters passed to the stored procedure.

The STCB is invisible to the Natural stored procedure called. The STCB is evaluated by the NDB server stub and stripped off the parameter list that is passed to the Natural stored procedure.

If the caller of a Natural stored procedure defined with PARAMETER STYLE (GENERAL or GENERAL WITH NULL) is a Natural program, the program should contain a CALLDBPROC statement with the keyword CALLMODE=NATURAL.

If the caller of the Natural stored procedure is **not** a Natural program, the caller has to set up the STCB for the DB2 CALL statement.

If an error occurs during the execution of a Natural stored procedure defined with PARAMETER STYLE either GENERAL or GENERAL WITH NULL, the error message text is returned to the STCB.

If the caller is a Natural program that uses CALLDBPROC and CALLMODE=NATURAL, the NDB runtime will wrap up the error text in the NAT3286 error message.

Example of PARAMETER STYLE GENERAL

In the Natural stored procedure, define the parameters as shown in the example program below:

```
DEFINE DATA PARAMETER
01 P1
01 P2


01 Pn  ???LOCAL


END-DEFINE
```

Example of GENERAL WITH NULL

In the Natural stored procedure, define the parameters as shown in the example program below:

```
DEFINE DATA PARAMETER
01 P1
01 P2


01 Pn  (???)
01 NULL-INDICATOR-ARRAY  (I2/1:n)
LOCAL


END-DEFINE
```

DB2SQL

A Natural stored procedure with PARAMETER STYLE DB2SQL receives the user parameters specified and, additionally, the following parameters:

- A NULL indicator for each parameter of the CALL statement,
- The SQLSTATE to be returned to DB2,
- The qualified name of the stored procedure,
- The specific name of the stored procedure,
- The SQL diagnostic string to be returned to DB2.

The parameters SQLSTATE, qualified and specific name, and SQL diagnostic string are defined in the Natural parameter data area (PDA) DB2SQL_P which is supplied in the Natural system library SYSDB2. If the option DBINFO is used, the DBINFO structure is passed to the Natural stored procedure. The DBINFO structure is described in the Natural PDA DBINFO_P supplied in the Natural system library SYSDB2.

???The STCB is required for Natural stored procedures defined with PARAMETER STYLE GENERAL (WITH NULL) in order to determine which subprogram to invoke from what library. However, the NDB server stub does not need the STCB in order to determine which subprogram to invoke from which library for Natural stored procedures defined with PARAMETER_STYLE DB2SQL.

Instead, the NDB server stub determines the subprogram and the library from the qualified and specific name of the stored procedure. The SCHEMA name is used as library name, and the procedure name is used as subprogram name.

The ROUTINEN subprogram is supplied in the Natural system library SYSDB2. This subprogram is used to access the DB2 catalog to determine the formats of the user parameters defined for the Natural stored procedure. After the formats have been determined, they are stored in the Natural buffer pool. During subsequent invocations of the Natural stored procedure, the formats are then retrieved from the Natural buffer pool. This requires that at least READS SQL DATA is specified for Natural stored procedures with PARAMETER STYLE DB2SQL.

The ROUTINEN subprogram is generated statically. The DBRM of ROUTINEN is bound as package in the COLLECTION SAGNDBROUTINENPACK. Before starting to access the DB2 catalog, the subprogram will save the CURRENT PACKAGESET and set SAGNDBROUTINENPACK to CURRENT PACKAGESET. After processing, the ROUTINEN subprogram will restore the CURRENT PACKAGESET saved.

If the caller of the Natural stored procedure with PARAMETER STYLE DB2SQL is a Natural program, the caller should use the CALLDBPROC statement with the specification CALLMODE=NONE, which is the default.

If the caller of the Natural stored procedure is **not** a Natural program, the caller does not have to set up the STCB for its DB2 CALL statement.

If a Natural runtime error occurs during the execution of a Natural stored procedure with PARAMETER STYLE DB2SQL, SQLSTATE is set to 38N99 and the diagnostic string contains the text of the Natural error message.

If an error occurs in the NDB server stub during the execution of the Natural stored procedure with PARAMETER STYLE DB2SQL, the SQLSTATE is set to 38S99 and the diagnostic string contains the text of the error message.

Example of DB2SQL

In the Natural stored procedure, define the parameters as shown in the example program below:

```
DEFINE DATA PARAMETER
01 P1
01 P2


01 PN
01 N1 (I2)
01 N2 (I2)


01 Nn (I2)       ???
PARAMETER USING DB2SQL_P
[ PARAMETER USING DBINFO_P ]   /*  only if DBINFO is defined
LOCAL


END-DEFINE
```

Writing a Natural Stored Procedure

Below are general instructions on what to consider when writing Natural stored procedures.

To write a Natural stored procedure

1. Determine the format and attributes of the parameters that are passed between the caller and the stored procedure.
   Consider creating a Natural PDA (parameter data area).
   Stored procedures do not support data groups and redefinition within their parameters.

2. Determine the PARAMETER STYLE of the stored procedure:
   GENERAL, GENERAL WITH NULL or DB2SQL.
   - If you use GENERAL WITH NULL, append the parameters to the Natural stored procedure by defining a NULL indicator array that contains a NULL indicator (I2) for each other parameter.
   - If you use DB2SQL, append the parameters of the Natural stored procedure by defining NULL indicators (one for each parameter), include the PDA DB2SQL_P and the PDA DBINFO_P (only with DBINFO specified), if desired.
     See also the relevant DB2 literature by IBM.
3. Decide which and how many result sets the stored procedure will return to the caller.
4. Code your stored procedure as a Natural subprogram.
   - **Returning result sets**
     To return result sets, code a SELECT statement with the WITH RETURN option.
     To return the whole result set, code an ESCAPE BOTTOM immediately after the SELECT.
     To return part of the result set code, an IF *COUNTER = 1 ESCAPE TOP END-IF immediately following the SELECT statement. This ensures that you do not process the first empty row that is returned by the SELECT WITH RETURN statement. To stop row processing, execute an ESCAPE BOTTOM statement.
     If you do not leave the processing loop initiated by the SELECT WITH RETURN via ESCAPE BOTTOM, the result set created is closed and nothing is returned to the caller.
   - **Attention when accessing other databases**
     You can access other databases (for instance Adabas) within a Natural stored procedure. However, you should keep in mind that your access to other databases is synchronized neither with the updates done by the caller of the stored procedure, nor with the updates done against DB2 within the stored procedure.
   - **NDB handling of COMMIT and ROLLBACK statements**
     DB2 does not allow a stored procedure to issue COMMIT or ROLLBACK statements (the execution of those statements puts the caller into a must-rollback state). Therefore, the NDB runtime handles those statements as follows when they are issued from a stored procedure:
     COMMIT against DB2 will be skipped.
     This allows the stored procedure to commit Adabas updates without getting a must-rollback state from DB2.
     ROLLBACK against DB2 will be skipped if it is created by Natural itself.
     ROLLBACK against DB2 will be executed if it is user-programmed.
     Thus, after a Natural error, the caller receives the Natural error information and not the unqualified must-rollback state. Additionally, this function ensures that, if the user program backs out the transaction, every database transaction of the stored procedure is backed out.
5. **For DB2 for OS/390 Version 5 and below:**
   Use the Procedure Maintenance function (see the relevant section) to define a stored procedure in the SYSIBM.SYSPROCEDURES table. To define a Natural stored procedure, enter the following data:

Column	**Data**
PROCEDURE	The name of your Natural subprogram representing your stored procedure.
LOADMOD	The name of your Natural for DB2 server stub module.
LINKAGE	Either blank or N as you decided at 2.
IBMREQD	N
PGM_TYPE	Depending on the specification of the MAIN parameter of the Natural for DB2 server stub module.
RESULT_SETS	As you decided at 3.
LANGUAGE	ASSEMBLER
PARMLIST	The description of the parameters you determined at 1. The first field of the PARMLIST **must** be defined as Natural internal parameter description STCB. It must be specified as a VARCHAR field with DATA INOUT and the following size: **274 + 13*N** where **N** is the number of parameters passed to the stored procedure (STCB not counted). If you have created a Natural PDA (parameter data area) for the stored procedure, you can easily derive the PARMLIST from your PDA by pressing PF5 on your PARMLIST screen of SYSDB2 and entering the library and the name of the PDA. In this case, the VARCHAR field for the parameter description is generated automatically.

For DB2 UDB for OS/390 Version 6 and above: Issue a CREATE PROCEDURE statement that defines your stored procedure, for example:

```
CREATE PROCEDURE <PROCEDURE>
  (INOUT  STCB       VARCHAR(274+13*N),
   INOUT  <PARM1>    <FORMAT>,
   INOUT  <PARM2>    <FORMAT>,
   INOUT  <PARM3>    <FORMAT>
   .
  )
  DYNAMIC RESULT SET <RESULT_SETS>
  EXTERNAL NAME <LOADMOD>
  LANGUAGE ASSEMBLE
  PROGRAM TYPE <PGM_TYPE>
  PARAMETER STYLE GENERAL <WITH NULLS depending on LINKAGE>;
```

The data specified in angle brackets (< >) correspond to the data listed in the table above, PARM1 - PARM3 and FORMAT depend on the call parameter list of the stored procedure.
See also Example Stored Procedure NDBPURGN, Member CR6PURGN below.

6. Code your Natural program invoking the stored procedure via the CALLDBPROC statement.

   Code the parameters in the CALLDBPROC statement in the same sequence as they are specified in the stored procedure. Define the parameters in the calling program in a format that is compatible with the format defined in the stored procedure.

   If you use result sets, specify a RESULT SETS clause in the CALLDBPROC statement followed by a number of result set locator variables of FORMAT (I4). The number of result set locator variables should be the same as the number or result sets created by the stored procedure. If you specify fewer than are created, some result sets are lost. If you specify more than are created, the remaining result set locator variables are lost. The sequence of locator variables corresponds to the sequence in which the result sets are created by the stored procedure.
   Keep in mind that the fields into which the result set rows are read have to correspond to the fields used in the

SELECT WITH RETURN statement that created the result set.

Security

DB2 provides an authorization ID for the execution of a Natural stored procedure to control access to non-SQL resources with an external security product, such as RACF. The NDB server stub uses this authorization ID to perform an implicit LOGON within the Natural session created for (???) the stored procedure. So, if the Natural stored procedure is to be executed in a Natural Security environment, ensure that the authorization ID used has been defined in the FSEC file.

The DB2 (???) authorization ID is the authorization ID for

- The address space in which the Natural stored procedure is running if SECURITY DB2 has been specified, or
- The process that invokes the Natural stored procedure if SECURITY USER has been specified, or
- The OWNER of the Natural (???) stored procedure if SECURITY DEFINER has been specified.

Example Stored Procedure NDBPURGN

This section describes the example stored procedure NDBPURGN, a Natural subprogram which purges Natural objects from the buffer pool used by the Natural stored procedures server.

Below is information on:

- Members of NDBPURGN
- Defining the Stored Procedure
- Stored Procedure Control Block (STCB)

Members of NDBPURGN

The example stored procedure NDBPURGN comprises the following text members which are stored in the library SYSDB2:

Member	Explanation
CR5PURGN	Input member for SYSDB2 ISQL. Contains SQL statements used to declare NDBPURGN in DB2 Version 5.
CR6PURGN	Input member for SYSDB2 ISQL. Contains SQL statements used to declare NDBPURGN in DB2 Version 6 and above.
NDBPURGP	The client (Natural) program which ● Requests the name of the program to be purged and the library where it resides, ● Invokes the stored procedure NDBPURGN and ● Reports the outcome of the request.
NDBPURGN	The stored procedure which purges objects from the buffer pool. NDBPURGN invokes the user exit USR0340N supplied in the library SYSEXT. Therefore, USR0340N must be available in the library defined as the steplib for the execution environment.

Defining the Stored Procedure NDBPURGN

To define the example stored procedure NDBPURGN

- Define the stored procedure in the DB2 catalog by using the SQL statements provided as text members CR5PURGN (for DB2 Version 5) and CR6PURGN (for DB2 Version 6).
- Specify the name of the Natural stored procedure stub (here: NDB41SRV) as LOADMOD (V5) or EXTERNAL NAME (V6). The Natural stored procedure stub is generated during the installation by assembling the NDBSTUB macro.
- As the first parameter, pass the internal Natural parameter STCB to the stored procedure.
  The STCB parameter is a VARCHAR field which contains information required to invoke the stored procedure in Natural:
  - The program name of the stored procedure and the library where it resides,
  - The description of the parameters passed to the stored procedure and
  - The error message created by Natural if the stored procedure fails during the execution.

  The STCB parameter is generated automatically by the CALLMODE=NATURAL clause of the CALLDBPROC statement and is removed from the parameters passed to the Natural stored procedure by the server stub. Thus, STCB is invisible to the caller and the stored procedure. However, if a non-Natural client intends to call a Natural stored procedure, the client has to pass the STCB parameter explicitly. See also Stored Procedure Control Block below.

Stored Procedure Control Block (STCB)

Below is the Stored Procedure Control Block (STBC) generated by the CALLMODE=NATURAL clause as generated by the stored procedure NDBPURGN before and after execution. Changed values are emphasized in boldface:

STCB before Execution:

```
004C82   0132F0F3  F0F6E2E3  C3C2F3F1  F040C8C7  *..0306STCB310 HG*    11097D42
004C92   D2404040  4040C8C7  D2404040  4040D5C4  *K     SAG     ND*    11097D52
004CA2   C2D7E4D9  C7D74040  40404040  4040F0F5  *BPURGP        05*    11097D62
004CB2   F7F0D5C4  C2D7E4D9  C7D5F0F0  F0F6F0F9  *70NDBPURGN000609*    11097D72
004CC2   F9F9F940  40404040  40404040  40404040  *999             *    11097D82
004CD2   40404040  40404040  40404040  40404040  *                *    11097D92
004CE2   40404040  40404040  40404040  40404040  *                *    11097DA2
004CF2   40404040  40404040  40404040  40404040  *                *    11097DB2
004D02   40404040  40404040  40404040  40404040  *                *    11097DC2
004D12   40404040  40404040  40404040  40404040  *                *    11097DD2
004D22   40404040  40404040  40404040  40404040  *                *    11097DE2
004D32   40404040  40404040  40404040  40404040  *                *    11097DF2
004D42   40404040  40404040  40404040  40404040  *                *    11097E02
004D52   40404040  40404040  40404040  40404040  *                *    11097E12
004D62   40404040  40404040  40404040  40404040  *                *    11097E22
004D72   40404040  40404040  40404040  40404040  *                *    11097E32
004D82   40404040  40404040  40404040  40404040  *                *    11097E42
004D92   40404040  D4C1F86B  F0D4C1F4  F06BF0D4  *    MA8,0MA40,0M*    11097E52
004DA2   C2F26BF0  D4C2F26B  F0D4C9F2  6BF0D4C9  *I2,0MI2,0MI2,0MI*    11097E62
004DB2   F26BF04B                                *2,0.            *    11097E72
```

STCB after Execution:

```
004C82   0132F0F3  F0F6E2E3  C3C2F3F1  F040C8C7  *..0306STCB310 HG*   11097D42
004C92   D2404040  4040C8C7  D2404040  4040D5C4  *K     SAG     ND*   11097D52
004CA2   C2D7E4D9  C7D74040  40404040  4040F0F5  *BPURGP        05*   11097D62
004CB2   F7F0D5C4  C2D7E4D9  C7D5F0F0  F0F6F0F0  *70NDBPURGN000600*   11097D72
004CC2   F0F0F040  40404040  40404040  40404040  *000             *   11097D82
004CD2   40404040  40404040  40404040  40404040  *                *   11097D92
004CE2   40404040  40404040  40404040  40404040  *                *   11097DA2
004CF2   40404040  40404040  40404040  40404040  *                *   11097DB2
004D02   40404040  40404040  40404040  40404040  *                *   11097DC2
004D12   40404040  40404040  40404040  40404040  *                *   11097DD2
004D22   40404040  40404040  40404040  40404040  *                *   11097DE2
004D32   40404040  40404040  40404040  40404040  *                *   11097DF2
004D42   40404040  40404040  40404040  40404040  *                *   11097E02
004D52   40404040  40404040  40404040  40404040  *                *   11097E12
004D62   40404040  40404040  40404040  40404040  *                *   11097E22
004D72   40404040  40404040  40404040  40404040  *                *   11097E32
004D82   40404040  40404040  40404040  40404040  *                *   11097E42
004D92   40404040  D4C1F86B  F0D4C1F4  F06BF0D4  *    MA8,0MA40,0M*   11097E52
004DA2   C2F26BF0  D4C2F26B  F0D4C9F2  6BF0D4C9  *I2,0MI2,0MI2,0MI*   11097E62
004DB2   F26BF04B                                *2,0.            *   11097E72
```